

# CSS: Cascading Style Sheets

Escuela Técnica Superior de Ingeniería de Telecomunicación  
Universidad Rey Juan Carlos

gsync-profes (arroba) gsync.urjc.es

Febrero de 2020



Las transparencias del tema “CSS - Hojas de estilo”  
están basadas en el libro de uniwebsidad.com  
disponible en <http://www.uniwebsidad.com/libros/css>  
Se ha pedido explícitamente autorización al autor original  
para realizar esta obra derivada con fines educativos.  
©Javier Eguiluz - uniwebsidad.com

Derivado a partir de material de  
Jesús M. González-Barahona y Gregorio Robles.  
El original está disponible en  
<http://cursosweb.github.io>  
Algunos derechos reservados.  
Este trabajo se distribuye bajo la licencia  
Creative Commons Attribution Share-Alike 4.0

# ¿Qué es CSS?

- CSS, *Cascading Style Sheets* es un lenguaje de hojas de estilos creado para **controlar el aspecto** o presentación de los documentos electrónicos definidos con HTML
- Es la mejor forma de **separar los contenidos y su presentación** y es imprescindible para crear páginas web complejas
  - Obliga a crear documentos HTML bien definidos y con significado completo (también llamados *documentos semánticos*)
  - Mejora la accesibilidad del documento
  - Reduce la complejidad de su mantenimiento
  - Permite adaptar el documento a dispositivos distintos, manteniendo el código HTML

# Antes del CSS

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
  ↪ charset=iso-8859-1"/>
  <title>Ejemplo de estilos sin CSS</title>
</head>

<body>
  <h1><font color="red" face="Arial" size="5">
    Titular de la página
  </font></h1>
  <p><font color="gray" face="Verdana" size="2">
    Un párrafo de texto no muy largo.
  </font></p>
</body>
</html>
```

# Con CSS

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Ejemplo de estilos con CSS</title>
  <style>
    h1 { color: red; font-family: Arial; font-size: large; }
    p  { color: gray; font-family: Verdana; font-size: medium; }
  </style>
</head>

<body>
  <h1>Titular de la página</h1>
  <p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```

# CSS en un documento HTML

Se pueden integrar instrucciones CSS de varias maneras en un documento HTML:

① Hoja de estilos externa

Con un elemento *link* en la cabecera, que apunte a un fichero .css que contenga las reglas

Recomendable para la mayoría de los casos

```
<link rel="stylesheet" type="text/css" href="estilos.css">
```

② Hoja de estilos interna

Escribir las reglas CSS dentro de un elemento *style*, en la cabecera del documento HTML

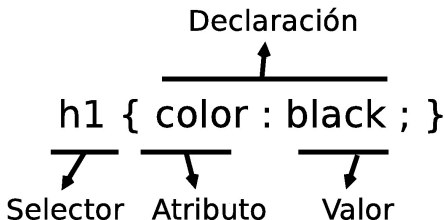
Adecuado para ejemplos sencillos

③ Reglas CSS insertadas directamente en los elementos HTML

Un atributo *style* directamente en cada elemento, cuyo valor sea la regla

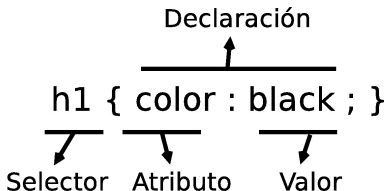
No recomendable

# Glosario Básico (I)



- **Regla:** cada uno de los estilos que componen una hoja de estilos CSS. Cada regla está compuesta de una parte de “selectores”, un símbolo de “llave de apertura” (`{`), otra parte denominada “declaración” y por último, un símbolo de “llave de cierre” (`}`).
- **Selector:** indica el elemento o elementos HTML a los que se aplica la regla CSS.

# Glosario Básico (y II)



- **Declaración:** especifica los estilos que se aplican a los elementos. Está compuesta por una o más atributos CSS.
- **Atributo:** característica que se modifica en el elemento seleccionado, como por ejemplo su tamaño de letra, su color de fondo, etc.
- **Valor:** establece el nuevo valor de la característica modificada en el elemento.

CSS 2.1 define 115 atributos, mientras que CSS 3 define 239



# Selectores

- A un mismo elemento HTML se le pueden aplicar varias reglas
- Cada regla puede aplicarse a un número ilimitado de elementos
- Cuando el selector de dos o más reglas CSS es idéntico, se pueden agrupar las declaraciones de las reglas para hacer las hojas de estilos más eficientes
- Cuando se establece el valor de un atributo CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de ese atributo

- CSS 2.1 incluye una docena de tipos diferentes de selectores, que permiten seleccionar de forma muy precisa elementos individuales o conjuntos de elementos dentro de una página web.
- Los selectores de CSS tienen mucha utilidad porque también son usados en jQuery

Resumen:

	Significado
-----	
(espacio)	descendiente (hijo, nieto...)
.	clase
,	OR
(concatenación)	AND
#	id

# Selectores básicos

- 1 Selector universal
- 2 Selector de tipo o etiqueta
- 3 Selector descendiente
- 4 Selector de clase
- 5 Selector de identidad

# Selector Universal

- No se utiliza habitualmente
- Generalmente es equivalente para poner estilo a `< body >`
- Se suele combinar con otros selectores y además, forma parte de algunos hacks muy utilizados

```
* {  
  margin: 0;  
  padding: 0;  
}
```

# Selector de tipo o etiqueta

- Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector
- Se pueden agrupar todas las reglas individuales en una sola regla con un selector múltiple. La coma significa *or*
- Buena práctica: agrupar los atributos comunes de varios elementos en una única regla CSS y posteriormente definir los atributos específicos de esos mismos elementos

```
h1, h2, h3 {  
  color: #8A8E27;  
  font-weight: normal;  
  font-family: Arial, Helvetica, sans-serif;  
}
```

```
h1 { font-size: 2em; }  
h2 { font-size: 1.5em; }  
h3 { font-size: 1.2em; }
```

# Selector descendiente

- Selecciona los elementos contenidos dentro de otros elementos.

Ejemplo: elementos span contenidos dentro de elementos p

```
p span { color: red; }  
[...]  
<p>  
  ...  
  <span>Texto1</span>  
  <a href="">...<span>Texto2</span></a>  
  ...  
</p>
```

Texto 1 evidentemente cumple la regla.

Texto 2 es un elemento span contenido dentro de un enlace contenido dentro de un elemento p. Por tanto, Texto 2 están contenido dentro de un p. Aunque no sea descendiente directo, es descendiente y la regla se aplica.

# Ejercicio

¿Qué elementos se seleccionarían con estos tipos de selectores?

- `p a span` em `{ text-decoration: underline; }`
- `p, a, span,` em `{ text-decoration: underline; }`
- `p a` `{ color: red; }`
- `p * a` `{ color: red; }`

# Selector de clase

- Se utiliza el atributo class de HTML sobre ese elemento para indicar directamente la regla CSS que se le debe aplicar
- Se crea en el archivo CSS una nueva regla llamada destacado con todos los estilos que se van a aplicar al elemento
- Se prefija el valor del atributo class con un punto (.)

```
.destacado { color: red; }  
[...]  
<p class="destacado">  
  Lorem ipsum dolor sit amet...  
</p>  
<p>Nunc sed lacus et  
  <a href="#" class="destacado">est adipiscing</a>  
</p>  
<p>Class aptent taciti <em class="destacado">sociosqu ad</em>  
</p>
```

Esta regla se aplica a cualquier elemento de clase *destacado*



El atributo clase puede tener varios valores, separados por espacios

```
<style>
.comentario {color : blue;}
.noticia {color : red;}
.obsoleto {text-decoration : line-through;}
</style>
[...]
<body>
  <p class="comentario"> Lorem ipsum dolor sit amet<p>
  <p class="comentario obsoleto"> consectetur adipisicing elit, </p>
  <p class="noticia "> sed do eiusmod tempor incididunt </p>
  <p class="noticia obsoleto"> ut labore et dolore magna aliqua.</p>
</body>
```

Ejemplo:

[http://ortuno.es/ej000\\_clases.html](http://ortuno.es/ej000_clases.html)

## Selector de clase más específico

- Combinando el selector de tipo y el selector de clase, se obtiene un selector mucho más específico.

```
p.destacado { color: red }  
[...]  
<p class="destacado">  
  Lorem ipsum dolor sit amet...  
</p>  
<p>Nunc sed lacus et  
  <a href="#" class="destacado">est adipiscing</a>  
</p>  
<p>Class aptent taciti <em class="destacado">sociosqu ad</em>  
</p>
```

Esta regla se aplica a los elementos de tipo párrafo, que además sean de clase *destacado*. (En este ejemplo, solo una vez)

```
.a.b {...}
```

Esta regla se aplica a los elementos de clase *a* que además sean de clase *b*

- Es equivalente a decir los elementos de clase *b* que además sean de clase *a* (las clases son atributos, y los atributos no tienen orden)
- Por supuesto, también se aplica a todos sus descendientes

[http://ortuno.es/concatenacion\\_clases.html](http://ortuno.es/concatenacion_clases.html)

# Ejercicio

¿Qué elementos se seleccionarían con estos tipos de selectores?

- `p.avisos { ... }`
- `p .avisos { ... }`
- `p, .avisos { ... }`
- `*.avisos { ... }`

# Selectores de identificador

- Aplica estilos CSS a un único elemento de la página
- El valor del atributo id no se puede repetir en dos elementos diferentes de una misma página

```
#destacado { color: red; }
```

```
<p>Primer párrafo</p>
```

```
<p id="destacado">Segundo párrafo</p>
```

```
<p>Tercer párrafo</p>
```

# Ejercicio

¿Qué elementos se seleccionarían con estos tipos de selectores?

- `p#aviso { ... }`
- `p #aviso { ... }`
- `p, #aviso { ... }`
- `*#aviso { ... }`

# Ejercicio: Combinación de selectores

¿Qué elementos se seleccionarían con estos tipos de selectores?

- `.aviso .especial { ... }`
- `div.aviso span.especial { ... }`
- `ul#menuPrincipal li.destacado a#inicio { ... }`

# Colisión de estilos (simplificado)

- 1 Cuanto más específico sea un selector, más importancia tiene su regla asociada.
- 2 A igual especificidad, se considera la última regla indicada.



# Unidades de medida

- Unidades absolutas
  - in, cm, mm, pt, pc
- Unidades relativas
  - em, ex, px
- Porcentajes

En general, se recomienda el uso de unidades relativas siempre que sea posible

Normalmente se utilizan

- Pixel y porcentajes  
Para definir el *layout* (la distribución) del documento. Esto es, la anchura de las columnas y de los elementos de las páginas
- em y porcentajes  
Para definir el tamaño de letra de los textos

# Especificación del color

Hay dos formas principales de indicar el color

- Mediante su nombre

Los 18 principales son:

red, cyan, blue, darkblue, lightblue, purple, yellow, lime, magenta, white, silver, gray/grey, black, orange, brown, maroon, green, olive

- Mediante su código hexadecimal

Se representa por con una almohadilla y 6 dígitos hexadecimales (dos dígitos para el rojo, dos dígitos para el verde y dos dígitos para el azul)

Ejemplos:

```
#FF0000 (red)
#F70D1A (ferrari red)
#FDEEF4 (pearl)
```

Para elegir un color, podemos usar cualquier herramienta online.  
Buscamos *html color picker*

Los principales atributos relacionados con el color son

- color
- background-color
- border-color

# Atributos relacionados con el texto

- Alineación

```
text-align: left | right | center | justify | initial | inherit;
```

- Subrayado

```
text-decoration: none | underline | overline | line-through |  
initial | inherit;
```

- Tamaño

```
font-size: medium | xx-small | x-small | small | large |  
x-large | xx-large | smaller | larger | (length) |  
initial | inherit;
```

- Estilo

```
font-style: normal | italic | oblique | initial | inherit;
```

# Atributos relacionados con los bordes

Los atributos del borde de una caja se especifican con:

- border-width
- border-style
- border-color

Puede abreviarse como simplemente *border*

- Ancho del borde

```
border-width: medium | thin | thick | (length) |  
             initial | inherit;
```

- Estilo del borde

```
border-style: none | hidden | dotted | dashed | solid | double |  
             groove | ridge | inset | outset | initial | inherit;
```

[https://www.w3schools.com/css/css\\_border.asp](https://www.w3schools.com/css/css_border.asp)

# Tipos de elementos (I)

El estándar HTML clasifica a todos sus elementos en dos grandes grupos:

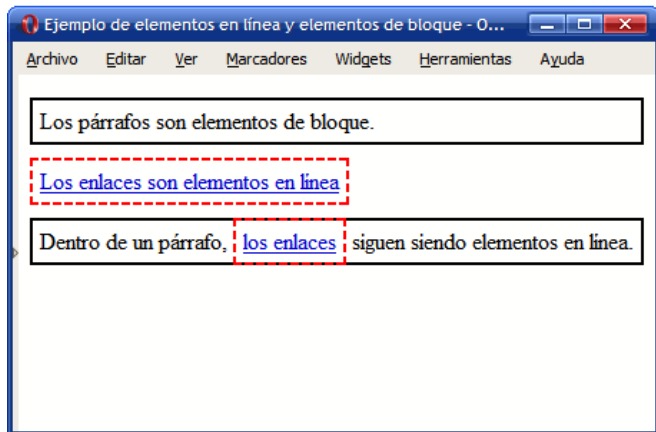
Elementos de línea:

- Los elementos en línea (“inline elements” en inglés) no empiezan necesariamente en nueva línea y sólo ocupan el espacio necesario para mostrar sus contenidos.

## Tipos de elementos (II)

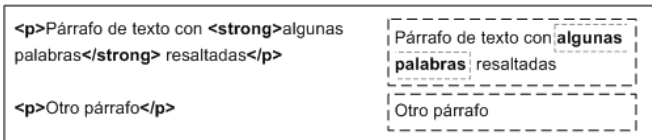
### Elementos de bloque:

- Los elementos de bloque (“block elements” en inglés) siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea



# El modelo de cajas

- Es el comportamiento de CSS que hace que todos los elementos de las páginas se representen mediante cajas rectangulares
- Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento





# El modelo de cajas (II)

- No son visibles a simple vista porque inicialmente no muestran ningún color de fondo ni ningún borde

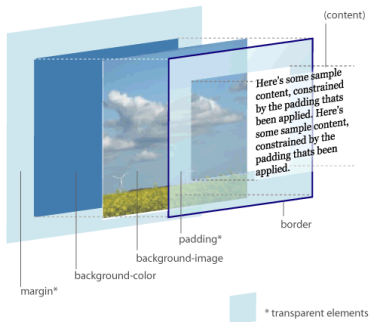


Ejemplo de <http://www.alistapart.com/> después de forzar a que todas las cajas muestren su borde

# El modelo de cajas (III)

- Los navegadores crean y colocan las cajas de forma automática, pero CSS permite modificar todas sus características. Cada una de las cajas está formada por seis partes, tal y como muestra la siguiente imagen:

THE CSS BOX MODEL HIERARCHY



(Esquema utilizado con permiso de <http://www.hicksdesign.co.uk/boxmodel/>)

# Partes que componen cada caja

- **Contenido** (content): se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
- **Relleno** (padding): espacio libre opcional existente entre el contenido y el borde.
- **Borde** (border): línea que encierra completamente el contenido y su relleno.
- **Imagen de fondo** (background image): imagen que se muestra por detrás del contenido y el espacio de relleno.
- **Color de fondo** (background color): color que se muestra por detrás del contenido y el espacio de relleno.
- **Margen** (margin): separación opcional existente entre la caja y el resto de cajas adyacentes.

# Margen, relleno, bordes y modelo de cajas (I)

- El margen, el relleno y los bordes establecidos a un elemento determinan la anchura y altura final del elemento

```
div {  
  width: 300px;  
  padding-left: 50px;  
  padding-right: 50px;  
  margin-left: 30px;  
  margin-right: 30px;  
  border: 10px solid black;  
}
```

Ejemplos:

<http://ortuno.es/cajas.html>

- Si indicamos 4 valores, se refieren a arriba, derecha, abajo, izquierda

```
margin: 8px 10px 8px 8px
```

- Si indicamos 3 valores, se refieren a arriba, ambos lados, abajo

```
margin: 8px 10px 8px
```

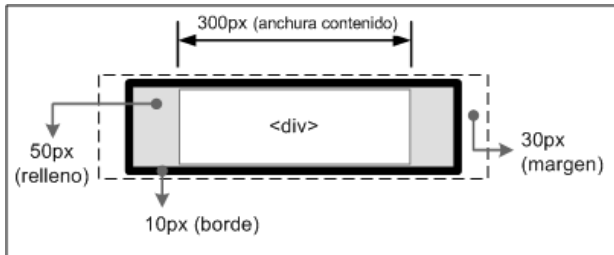
- Si indicamos 2 valores, se refieren a márgenes verticales, márgenes horizontales

```
margin: 8px 10px
```

- Si indicamos 1 valor, se aplica a los 4 márgenes (arriba, derecha, abajo, izquierda, todos iguales)

```
margin: 8px
```

# Margen, relleno, bordes y modelo de cajas (y II)



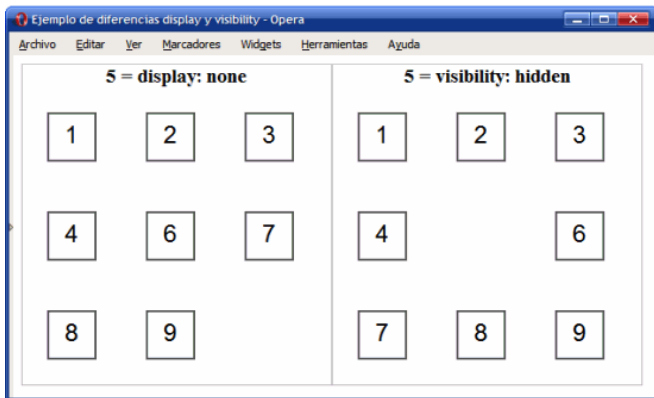
De esta forma, la anchura del elemento en pantalla sería igual a la suma de la anchura original, los márgenes, los bordes y los rellenos:

$$30\text{px} + 10\text{px} + 50\text{px} + 300\text{px} + 50\text{px} + 10\text{px} + 30\text{px} = 480 \text{ píxel}$$

# Visualización

- CSS define otros cuatro atributos para controlar su visualización: `display`, `visibility`, `overflow` y `z-index`.
- El atributo `display` permite ocultar completamente un elemento haciendo que desaparezca de la página. Como el elemento oculto no se muestra, el resto de elementos de la página se mueven para ocupar su lugar.
- El atributo `display` también permite modificar el comportamiento de un elemento a bloque (`block`) o en línea (`inline`).
- El atributo `visibility` permite hacer invisible un elemento, lo que significa que el navegador crea la caja del elemento pero no la muestra. En este caso, el resto de elementos de la página no modifican su posición, ya que aunque la caja no se ve, sigue ocupando sitio.

# Diferencias entre display y visibility





# Otros atributos

CSS tiene muchos otros atributos que no veremos aquí

- Normalmente los programadores no sabemos hacer buenos diseños gráficos, no es recomendable empeñarse en configurar el CSS a bajo nivel
- Si realmente quieres usar otros atributos, recuerda que *Google es tu amigo*

Ejemplo: Centrar una imagen

[https://gsyc.urjc.es/~mortuno/imagen\\_centrada.html](https://gsyc.urjc.es/~mortuno/imagen_centrada.html)